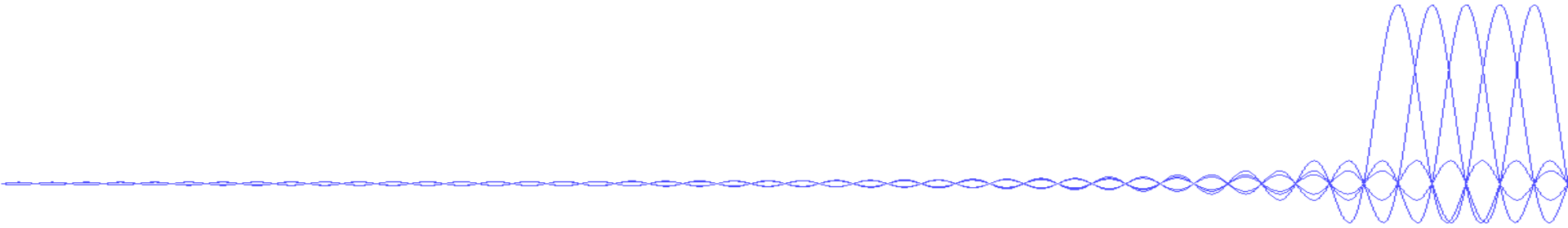
COMPUTER ENGINEERING

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

HỆ ĐIỀU HÀNH
Chương 9
Hệ điều hành Linux

17-Mar-20



COMPUTER ENGINEERING

Câu hỏi ôn tập chương 8

1. Tại sao cần phải có bộ nhớ ảo?
2. Có bao nhiêu kỹ thuật cài đặt bộ nhớ ảo? Mô tả sơ lượt các kỹ thuật đó?
3. Các bước thực hiện kỹ thuật phân trang theo yêu cầu?
4. Mô tả các giải thuật thay thế trang FIFO, OPT, LRU?
5. Giải pháp tập làm việc hoạt động như thế nào?

Bài tập chương 8

Xét chuỗi truy xuất bộ nhớ sau:

1, 2, 3, 4, 2, 1, 5, 6, 2, 1, 2, 3, 7, 6, 3, 2, 1, 2, 3, 6

Có bao nhiêu lỗi trang xảy ra khi sử dụng các thuật toán thay thế sau đây, giả sử có lần lượt là 2, 3, 4, 5 khung trang.

a. LRU
b. FIFO
c. Chiến lược tối ưu (OPT)

COMPUTER ENGINEERING

Mục tiêu chương 9

- Hiểu được các kiến thức cơ bản về hệ điều hành Linux.
- Phân tích, so sánh, đánh giá các kiến thức đã học và cách thức áp dụng các kiến thức đó vào hệ điều hành Linux.

COMPUTER ENGINEERING

Nội dung chương 9

- Hệ điều hành Linux:
  - Lịch sử phát triển
  - Nguyên tắc thiết kế
  - Các thành phần chính
  - Quản lý tiến trình
  - Cách thức xử lý định thời
  - Giao tiếp liên tiến trình
  - Quản lý bộ nhớ trên Linux.

COMPUTER ENGINEERING

Lịch sử phát triển của Linux

- Linux là một hệ điều hành hiện đại, miễn phí dựa trên UNIX.
- Nhân Linux bắt đầu được phát triển bởi Linus Torvalds từ 1991, với mục đích ban đầu để tương thích với UNIX, được phát hành dưới dạng mã nguồn mở.
- Linux được phát triển và duy trì bởi nhiều người dùng trên thế giới.
- Linux được thiết kế để hoạt động hiệu quả trên PC và nhiều nền tảng phần cứng khác.
- Linux có nhiều bản phân phối khác nhau bao gồm nhân, các ứng dụng và công cụ quản lý.

COMPUTER ENGINEERING

Nhân Linux

- Phiên bản 0.01 (5/1991) không có kết nối mạng, chỉ chạy trên PC với bộ xử lý Intel 80386.
- Phiên bản 1.0 (3/1994) bổ sung nhiều chức năng mới:
  - Hỗ trợ giao thức TCP/IP
  - Giao tiếp socket tương thích BSD
  - Cải thiện hệ thống tập tin
  - Hỗ trợ thêm nhiều phần cứng
- Phiên bản 1.2 (3/1995) là phiên bản cuối cùng chỉ dành cho PC.
- Cách thức đánh số phiên bản: số lẻ là các phiên bản phát triển, số chẵn là các phiên bản chính (production).

COMPUTER ENGINEERING

Linux 2.0

- Linux 2.0 được phát hành vào 6/1996 với hai cải tiến lớn:
  - Hỗ trợ nhiều kiến trúc, bao gồm cả 64 bit Alpha
  - Hỗ trợ các kiến trúc đa bộ xử lý
- Linux 2.0 hoạt động trên nhiều hệ thống khác nhau: Sun Sparc, PC, PowerMac, …
- Các phiên bản 2.4 và 2.6 tiếp tục tăng cường hỗ trợ SMP, cải thiện hệ thống quản lý bộ nhớ với sự hỗ trợ bộ nhớ 64 bit.
- Linux 3.0 phát hành vào 7/2011 với sự cải thiện khả năng ảo hóa, quản lý bộ nhớ và định thời.

COMPUTER ENGINEERING

Hệ thống Linux

- Linux sử dụng nhiều công cụ được phát triển bởi hệ điều hành Berkeley BSD, MIT’s X Window System và dự án Free Software Foundation's GNU.
- Hệ thống thư viện chính được bắt đầu từ dự án GNU với nhiều cải tiến được thực hiện bởi cộng đồng Linux.
- Các công cụ quản lý mạng trên Linux được kế thừa từ 4.3BSD.

COMPUTER ENGINEERING

Các bản phân phối Linux

- Các bản phân phối (distribution) là tập hợp các gói phần mềm đã được biên dịch và tiêu chuẩn hóa, bao gồm hệ thống Linux cơ bản, hệ thống cài đặt, các công cụ quản lý và các gói công cụ UNIX phổ biến.
- Các bản phân phối phổ biến hiện nay là RedHat (thương mại) và Debian (miễn phí). Một số bản phân phối thường gặp khác là Canonical và SuSE.
- Định dạng gói RPM có tính tương thích cao và được sử dụng bởi nhiều bản phân phối.

COMPUTER ENGINEERING

Giấy phép Linux

- Nhân Linux được phân phối dưới giấy phép GNU General Public License (GPL).
  - Nó không nằm trong public domain, bởi không phải tất cả các quyền đều từ bỏ.
- Bất kỳ ai sử dụng Linux, hoặc tạo ra phiên bản phái sinh của Linux, không được để sản phẩm đó là độc quyền, phần mềm được phát hành dưới giấy phép GPL không thể được tái phân phối chỉ dưới dạng nhị phân.
  - Có thể bán các bản phân phối, nhưng phải cung cấp kèm mã nguồn.

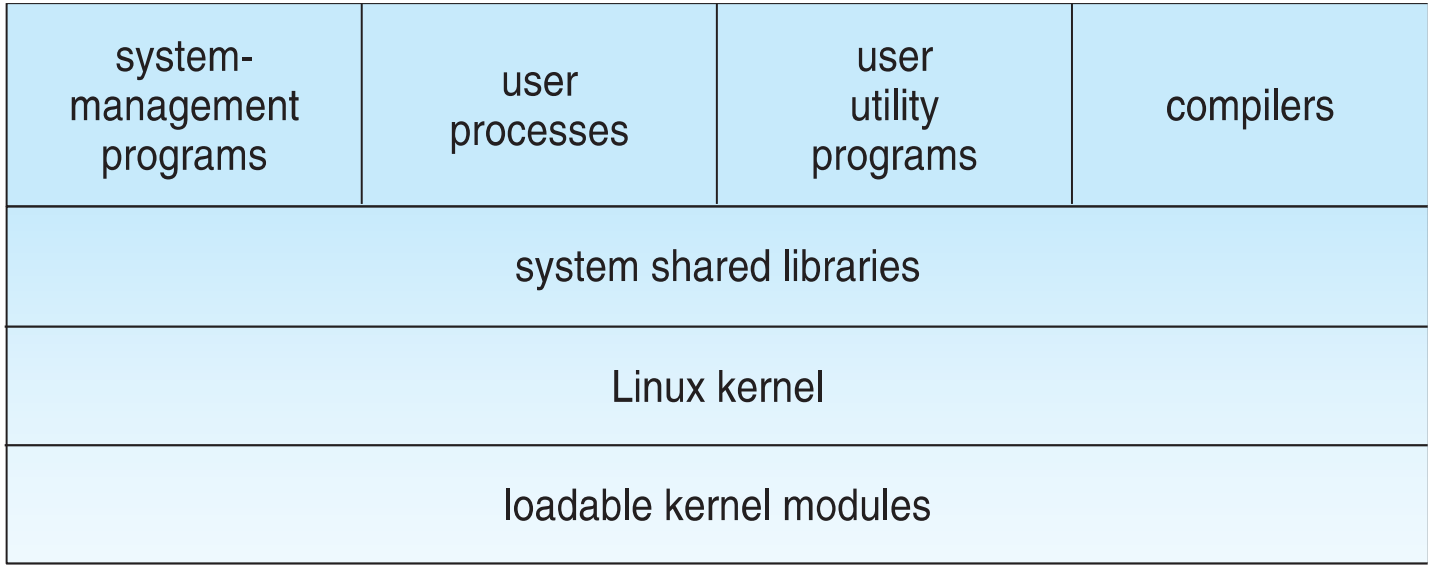
COMPUTER ENGINEERING

Nguyên tắc thiết kế

- Linux là một hệ thống nhiều người dùng, đa tác vụ với một tập đầy đủ các công cụ tương thích với UNIX.
- Các mục tiêu thiết kế chính là tốc độ, hiệu quả và tiêu chuẩn hóa.
- Linux được thiết kế để đáp ứng các yêu cầu POSIX liên quan.
- Hệ thống tập tin Linux tuân thủ theo UNIX. Linux cũng cài đặt đầy đủ mô hình mạng tiêu chuẩn của UNIX.
- Giao diện lập trình Linux tuân thủ theo SVR4 UNIX.

COMPUTER ENGINEERING

Các thành phần của hệ thống Linux



COMPUTER ENGINEERING

Các thành phần của hệ thống Linux

- Như phần lớn các cài đặt UNIX, hệ thống Linux bao gồm 3 thành phần chính: nhân, thư viện hệ thống và công cụ hệ thống.
- Nhân chịu trách nhiệm duy trì các hoạt động ở mức trừu tượng của hệ điều hành, như bộ nhớ ảo và tiến trình.
  - Các mã của nhân thực thi ở kernel mode với đầy đủ quyền truy xuất đến tài nguyên vật lý của máy tính.
  - Tất cả các mã của nhân và cấu trúc dữ liệu được lưu trên cùng một không gian địa chỉ.

COMPUTER ENGINEERING

Các thành phần của hệ thống Linux

- Các thư viện hệ thống định nghĩa một tập chuẩn các hàm mà thông qua đó các ứng dụng có thể tương tác với nhân. Các thư viện này cài đặt nhiều chức năng của hệ điều hành mà không cần đầy đủ quyền (privileges) như các mã của nhân.
- Các công cụ hệ thống là các chương trình thực hiện các chức năng quản lý cụ thể.

COMPUTER ENGINEERING

Các module nhân

- Các phần mã của nhân có thể biên dịch, nạp và gỡ độc lập với phần còn lại của nhân.
- Một module nhân có thể cài đặt một trình điều khiển thiết bị, một hệ thống tập tin hoặc một giao thức mạng.
- Các module nhân cho phép thiết lập một hệ thống Linux với một nhân Linux tiêu chuẩn tối thiểu mà không cần bất cứ trình điều khiển thiết bị đi kèm.
- Các giao thức của module có thể cho phép bên thứ ba viết và phân phối trình điều khiển thiết bị hoặc hệ thống tập tin của họ, vốn không thể phân phối dưới giấy phép GPL.

Các module nhân

- Linux hỗ trợ 4 loại module sau:
  - module-management system: cho phép module nạp vào bộ nhớ và giao tiếp với phần còn lại của nhân.
  - module loader and unloader: là các công cụ ở user mode, làm việc với module-management system để nạp một module vào bộ nhớ.
  - driver-registration system: cho phép các module thông báo với phần còn lại của nhân là có một trình điều khiển mới đã sẵn sàng.
  - conflict-resolutio mechanism: cho phép các trình điều khiển khác nhau chiếm lấy tài nguyên máy tính và bảo vệ tài nguyên này khỏi việc truy xuất không phù hợp từ trình điều khiển khác.

COMPUTER ENGINEERING

Quản lý tiến trình

- Hệ thống quản lý tiến trình của UNIX phân chia việc tạo tiến trình và chạy một chương trình mới thành hai thao tác riêng biệt.
  - System call fork() tạo ra một tiến trình mới.
  - Một chương trình mới được chạy sau khi gọi exec()
- Trên UNIX, một tiến trình bao gồm luôn tất cả thông tin mà hệ điều hành phải lưu trữ để lưu vết ngữ cảnh của một thao tác thực thi của một chương trình đơn.
- Trên Linux, các thuộc tính của tiến trình được chia thành 3 nhóm: định danh của tiến trình, môi trường và ngữ cảnh.

COMPUTER ENGINEERING

Định thời

- Trên Linux, định thời không chỉ việc chạy và tạm dừng các tiến trình, mà còn bao gồm việc thực thi nhiều tác vụ trong nhân.
- Các tác vụ trong nhân bao gồm các tác vụ được yêu cầu bởi tiến trình đang chạy và các tác vụ thực thi nội tại trong trình điều khiển thiết bị.
- Phiên bản 2.5 giới thiệu và sử dụng bộ định thời O(1) dựa trên độ ưu tiên và chế độ quyết định trưng dụng.
- Phiên bản 2.6 sử dụng Completely Fair Scheduler (CFS).

COMPUTER ENGINEERING

Giao tiếp liên tiến trình

- Tương tự UNIX, Linux thông báo cho các tiến trình có một sự kiện đã xảy ra thông qua các signal.
- Số lượng signal là giới hạn và chúng không chứa thông tin.
- Signal có thể được tạo ra bởi tiến trình hoặc nhân. Tuy nhiên, nhân không dùng signal để giao tiếp với tiến trình đang chạy ở kernel mode, thay vào đó, nó sử dụng các trạng thái định thời và cấu trúc *wait_queue*.

COMPUTER ENGINEERING

Chuyển dữ liệu giữa các tiến trình

- Pipe: Cho phép tiến trình con kế thừa một kênh giao tiếp từ tiến trình cha của nó. Dữ liệu ghi ở một đầu của pipe có thể đọc ở đầu còn lại.
- Network: UNIX cung cấp một tập các công cụ mạng để gửi dữ liệu đến các tiến trình cục bộ và từ xa.
- Shared memory: Cho phép giao tiếp dữ liệu một cách nhanh chóng. Dữ liệu được ghi bởi một tiến trình vào một vùng nhớ chia sẻ có thể được đọc ngay lập tức bởi một tiến trình khác nếu nó đã ánh xạ vùng nhớ đó vào không gian nhớ của nó.
  - Để đạt được tính đồng bộ, shared memory cần được sử dụng kết hợp với các phương thức giao tiếp khác.

COMPUTER ENGINEERING

Quản lý bộ nhớ

- Quản lý bộ nhớ trên Linux gồm 2 thành phần:
  - Cấp phát và giải phóng bộ nhớ vật lý – trang, nhóm các trang và các khối RAM
  - Xử lý bộ nhớ ảo – ánh xạ bộ nhớ vào không gian địa chỉ của các tiến trình đang chạy

Quản lý bộ nhớ

- Quản lý bộ nhớ vật lý:
  - Tùy thuộc vào đặc điểm phần cứng, Linux chia bộ nhớ vật lý thành 4 vùng: ZONE DMA, ZONE DMA32, ZONE NORMAL, ZONE HIGHMEM.
  - Mỗi vùng có page allocator riêng, chịu trách nhiệm cấp phát và giải phóng tất cả các trang vật lý, cũng như cấp phát một dãy các trang vật lý liên tiếp khi được yêu cầu.
  - Page allocator sử dụng giải thuật buddy heap để lưu vết các trang vật lý khả dụng.
    - Mỗi vùng nhớ cấp phát có một vùng nhớ liền kề - buddy.
    - Nếu hai vùng nhớ liền kề đều trống, chúng được kết hợp thành một vùng nhớ lớn hơn.
    - Vùng nhớ trống kích thước lớn có thể được chia thành 2 vùng nhớ để đáp ứng các yêu cầu cấp phát nhỏ.

COMPUTER ENGINEERING

Bộ nhớ ảo

- Hệ thống bộ nhớ ảo của Linux duy trì không gian địa chỉ cho mỗi tiến trình. Nó tạo các trang ảo theo yêu cầu và quản lý việc nạp các trang ảo từ đĩa cũng như di chuyển chúng trở về đĩa khi được yêu cầu.
- Không gian địa chỉ của một tiến trình có thể chia thành 2 view riêng biệt:
  - Logical view: Mô tả các hướng dẫn liên quan đến việc bố trí không gian địa chỉ. Không gian địa chỉ chứa một tập các vùng nhớ không chồng lấn nhau, mỗi vùng nhớ là một tập các trang nhớ liên tục.
  - Physical view: Được lưu trữ trên bảng trang của tiến trình và được quản lý bởi một tập các routines.

COMPUTER ENGINEERING

Bộ nhớ ảo

- Vùng nhớ ảo được đặc trưng bởi:
  - Backing store: nơi lấy ra các trang của vùng nhớ, thường là một tập tin hoặc không có gì (demand-zero memory).
  - Cơ chế đối với thao tác ghi: chia sẻ trang hoặc copy-on-write.
- Nhân tạo một không gian địa chỉ mới khi:
  - Một tiến trình thực thi một chương trình mới qua system call exec()
  - Trong khi tạo một tiến trình mới với system call fork()

COMPUTER ENGINEERING

Bộ nhớ ảo

- Khi thực thi một chương trình mới, tiến trình được cung cấp một vùng nhớ ảo mới, hoàn toàn trống. Routine chịu trách nhiệm nạp chương trình sẽ nạp đầy (populate) không gian địa chỉ này với các vùng nhớ ảo.
- Quá trình tạo một tiến trình mới với fork() sẽ tạo ra một bản sao đầy đủ của không gian địa chỉ ảo của tiến trình hiện tại.
  - Nhân sẽ sao chép các thông tin cấu trúc bộ nhớ ảo của cha, sau đó tạo ra một tập các bảng trang cho con.
  - Các bảng trang của cha được sao chép trực tiếp cho con.
  - Kết thúc fork(), cha và con cùng chia sẻ cùng số trang nhớ vật lý trong không gian địa chỉ của chúng.

COMPUTER ENGINEERING

Hoán vị và phân trang

- Hệ thống bộ nhớ ảo phân trang cần phải tái định vị các trang nhớ của bộ nhớ vật lý được đưa ra xuống đĩa khi cần thêm bộ nhớ cho một tác vụ nào đó.
- Hệ thống phân trang được chia thành 2 phần:
  - Giải thuật pageout-policy: Quyết định trang nào sẽ được ghi ra đĩa và khi nào thì thực hiện.
  - Paging mechanism: Thực hiện việc di chuyển, đồng thời cũng phụ trách việc đem dữ liệu trang trở lại bộ nhớ vật lý.
    - Trang có thể được đưa đến thiết bị khác, phân vùng khác hoặc tập tin.
    - Sử dụng giải thuật next – fit để ghi các trang liên tục.

COMPUTER ENGINEERING

Tóm tắt nội dung buổi học

- Hệ điều hành Linux:
  - Lịch sử phát triển
  - Nguyên tắc thiết kế
  - Các thành phần chính
  - Quản lý tiến trình
  - Cách thức xử lý định thời
  - Giao tiếp liên tiến trình
  - Quản lý bộ nhớ trên Linux.

COMPUTER ENGINEERING

Câu hỏi ôn tập

- Nguyên tắc thiết kế của hệ điều hành Linux?
- Các thành phần của hệ điều hành Linux?
- Các loại module nhân trong Linux?
- Linux quản lý tiến trình như thế nào? Có điểm gì khác so với kiến thức đã học?
- Linux quản lý bộ nhớ như thế nào? Có điểm gì khác so với kiến thức đã học?
- Trình bày bộ nhớ ảo trên Linux? Có điểm gì khác so với kiến thức đã học?

COMPUTER ENGINEERING

UIT
TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

THẢO LUẬN